

2001, whose inventors are Peter Geiger, Manuel J. Alvarez II, and Thomas A. Dye, is hereby incorporated by reference in its entirety as though fully and completely set forth herein.

*A1*  
*Cond*  
U.S. patent application Serial no. 10/044,785 titled "System And Method For Generating Optimally Compressed Data From A Plurality Of Data Compression / Decompression Engines Implementing Different Data Compression Algorithms", and filed January 11, 2002, currently herewith, whose inventors are Peter D. Geiger; Manuel J. Alvarez II; Thomas A. Dye, is hereby incorporated by reference in its entirety as though fully and completely set forth herein.

*In The Claims*

CLEAN COPY OF PENDING CLAIMS

Please amend claims 1, 18, 32, 35, 40, 45, 50, 52, 57, 63, 65, 76, 82, 85, 86, 93, 100, 107, 111, 113, 116, and 119 as follows.

The following is a clean version of the entire set of pending claims (unamended claims appear in smaller print).

- A2*
1. (Amended) A data compression system comprising:  
a plurality of parallel compression engines, wherein each of the plurality of parallel compression engines operates independently and implements a parallel data compression algorithm;  
wherein each of the plurality of parallel compression engines is operable to:  
receive a different respective portion of uncompressed data; and  
compress the different respective portion of the uncompressed data using the parallel data compression algorithm to produce a respective compressed portion of the uncompressed data; and  
output the respective compressed portion;

wherein the plurality of parallel compression engines are configured to perform said compression in a parallel fashion to produce a plurality of respective compressed portions of the uncompressed data.

2. The data compression system of claim 1, wherein, in performing said compression in a parallel fashion, the plurality of parallel compression engines operate concurrently to compress the different respective portions of the uncompressed data to produce the compressed portions of the uncompressed data.

3. The data compression system of claim 1, wherein the respective compressed portions output from the plurality of parallel compression engines are combinable to form compressed data corresponding to the uncompressed data.

4. The data compression system of claim 1, wherein each of the plurality of parallel compression engines implements a parallel lossless data compression algorithm.

5. The data compression system of claim 1, wherein each of the plurality of parallel compression engines implements a parallel statistical data compression algorithm.

6. The data compression system of claim 1, wherein each of the plurality of parallel compression engines implements a parallel dictionary-based data compression algorithm.

7. The data compression system of claim 6, wherein each of the plurality of parallel compression engines implements a parallel data compression algorithm based on a Lempel-Ziv (LZ) algorithm.

8. The data compression system of claim 6, wherein the uncompressed data comprises a plurality of symbols, wherein each of the plurality of parallel compression engines is operable to compare each of a plurality of received symbols with each of a plurality of entries in a history table concurrently.

9. The data compression system of claim 6, wherein each of the plurality of parallel compression engines comprises:  
an input for receiving the different respective portion of the uncompressed data, wherein the uncompressed data comprises a plurality of symbols, wherein the plurality of symbols includes a first symbol, a last symbol, and one or more middle symbols;  
a history table comprising entries, wherein each entry comprises at least one symbol;  
a plurality of comparators for comparing the plurality of symbols with entries in the history table, wherein the plurality of comparators are operable to compare each of the plurality of symbols with each entry in the history table concurrently, wherein the plurality of comparators produce compare results;

match information logic coupled to the plurality of comparators for determining match information for each of the plurality of symbols based on the compare results, wherein the match information logic is operable to determine if a contiguous match occurs for one or more of the one or more middle symbols that does not involve a match with either the first symbol or the last symbol; and

an output coupled to the match information logic for outputting compressed data in response to the match information.

10. The data compression system of claim 1, wherein the parallel data compression algorithm is based on one of an LZSS algorithm, an LZ77 algorithm, an LZ78 algorithm, an LZW algorithm, an LZRW1 algorithm, a Run Length Encoding (RLE) algorithm, a Predictive Encoding algorithm, a Huffman coding algorithm, an Arithmetic coding algorithm and a Differential compression algorithm.

11. The data compression system of claim 1, further comprising:  
a plurality of parallel decompression engines, wherein each of the plurality of parallel decompression engines implements a parallel data decompression algorithm;  
wherein each of the plurality of parallel compression engines is operable to:  
receive a different respective portion of compressed data; and  
decompress the different respective portion of the compressed data using the parallel data decompression algorithm to produce a respective uncompressed portion of the compressed data; and  
output the respective uncompressed portion;  
wherein the plurality of parallel decompression engines are configured to perform said decompression in a parallel fashion to produce a plurality of respective uncompressed portions of the compressed data.

12. The data compression system of claim 11,  
wherein, in performing said decompression in a parallel fashion, the plurality of parallel decompression engines operate concurrently to decompress the different respective portions of the compressed data to produce the uncompressed portions of the compressed data.

13. The data compression system of claim 11,  
wherein the respective uncompressed portions output from the plurality of parallel decompression engines are combinable to form uncompressed data corresponding to the compressed data.

14. The data compression system of claim 11, wherein each of the plurality of parallel decompression engines implements a parallel lossless data decompression algorithm.

15. The data compression system of claim 11, wherein each of the plurality of parallel decompression engines implements a parallel statistical data decompression algorithm.

16. The data compression system of claim 11, wherein each of the plurality of parallel decompression engines implements a parallel dictionary-based data decompression algorithm.

17. The data compression system of claim 11, wherein the compressed data comprises a compressed representation of uncompressed data, wherein the uncompressed data has a plurality of symbols, wherein each of the plurality of parallel decompression engines is operable to:

- receive the compressed data, wherein the compressed data comprises tokens each describing one or more of the symbols in the uncompressed data;
- examine a plurality of tokens from the compressed data in parallel in a current decompression cycle;
- and
- generate the uncompressed data comprising the plurality of symbols in response to said examining.

A2

18. (Amended) A data compression system comprising:

- a plurality of parallel compression engines, wherein each of the plurality of parallel compression engines operates independently and implements a parallel data compression algorithm;
- first logic coupled to the plurality of parallel compression engines and configured to:
  - receive uncompressed first data; and
  - provide a different respective portion of the uncompressed first data to each of the plurality of parallel compression engines;
- wherein each of the plurality of parallel compression engines is configured to:
  - compress the different respective portion of the uncompressed first data using the parallel data compression algorithm to produce a compressed portion of the first uncompressed data; and
  - output the compressed portion of the first uncompressed data;
- wherein the plurality of parallel compression engines are configured to perform said compression in a parallel fashion to produce a plurality of compressed portions of the first uncompressed data.

19. The data compression system of claim 18, wherein, in performing said compression in a parallel fashion, the plurality of parallel compression engines operate concurrently to compress the different respective portions of the uncompressed first data to produce the compressed portions of the first uncompressed data.

20. The data compression system of claim 18, further comprising:  
second logic coupled to the plurality of parallel compression engines and configured to:  
receive the plurality of compressed portions of the data; and  
merge the plurality of compressed portions of the first data to produce compressed first data.
21. The data compression system of claim 18, wherein each of the plurality of parallel compression engines implements a parallel lossless data compression algorithm.
22. The data compression system of claim 18, wherein each of the plurality of parallel compression engines implements a parallel statistical data compression algorithm.
23. The data compression system of claim 18, wherein each of the plurality of parallel compression engines implements a parallel dictionary-based data compression algorithm.
24. The data compression system of claim 23, wherein each of the plurality of parallel compression engines implements a parallel data compression algorithm based on a Lempel-Ziv (LZ) algorithm.
25. The data compression system of claim 23, wherein the uncompressed data comprises a plurality of symbols, wherein each of the plurality of parallel compression engines is operable to compare each of a plurality of received symbols with each of a plurality of entries in a history table concurrently.
26. The data compression system of claim 23,  
wherein each of the plurality of parallel compression engines comprises:  
an input for receiving the different respective portion of the uncompressed first data, wherein the  
uncompressed first data comprises a plurality of symbols, wherein the plurality of symbols  
includes a first symbol, a last symbol, and one or more middle symbols;  
a history table comprising entries, wherein each entry comprises at least one symbol;  
a plurality of comparators for comparing the plurality of symbols with entries in the history table,  
wherein the plurality of comparators are operable to compare each of the plurality of symbols  
with each entry in the history table concurrently, wherein the plurality of comparators produce  
compare results;  
match information logic coupled to the plurality of comparators for determining match information for  
each of the plurality of symbols based on the compare results, wherein the match information  
logic is operable to determine if a contiguous match occurs for one or more of the one or more  
middle symbols that does not involve a match with either the first symbol or the last symbol;  
and  
an output coupled to the match information logic for outputting compressed data in response to the  
match information.

27. The data compression system of claim 18, wherein the parallel data compression algorithm is based on a serial dictionary-based data compression algorithm.

28. The data compression system of claim 18, wherein the parallel data compression algorithm is based on one of an LZSS algorithm, an LZ77 algorithm, an LZ78 algorithm, an LZW algorithm, an LZRW1 algorithm, a Run Length Encoding (RLE) algorithm, a Predictive Encoding algorithm, a Huffman coding algorithm, an Arithmetic coding algorithm and a Differential compression algorithm.

A<sup>2</sup>

29. The data compression system of claim 18, further comprising:  
a plurality of parallel decompression engines, wherein each of the plurality of parallel decompression engines implements a parallel data decompression algorithm;  
third logic coupled to the plurality of parallel decompression engines and configured to:  
receive compressed second data; and  
provide a different respective portion of the compressed second data to each of the plurality of parallel decompression engines;  
wherein each of the plurality of parallel decompression engines is configured to:  
decompress the different respective portion of the compressed second data to produce an uncompressed portion of the second data; and  
output the uncompressed portion of the compressed second data;  
wherein the plurality of parallel decompression engines are configured to perform said decompression in a parallel fashion to produce a plurality of uncompressed portions of the compressed second data.

30. The data compression system of claim 29,  
wherein, in performing said decompression in a parallel fashion, the plurality of parallel decompression engines operate concurrently to produce the plurality of uncompressed portions of the compressed second data.

31. The data compression system of claim 29, further comprising:  
fourth logic coupled to the plurality of parallel decompression engines and configured to:  
receive the plurality of uncompressed portions of the compressed second data; and  
merge the plurality of uncompressed portions of the compressed second data to produce uncompressed second data.

32. (Amended) A data compression system comprising:  
a plurality of compression engines, wherein each of the plurality of compression engines operates independently and implements a parallel data compression algorithm;  
first logic coupled to the plurality of compression engines and configured to:  
receive uncompressed data;

provide a different portion of the uncompressed data to each of the plurality of compression engines;

wherein each of the plurality of compression engines is configured to compress a received uncompressed portion of the data to produce a compressed portion of the data, wherein, in said compressing, each of the plurality of compression engines is configured to:

maintain a history table comprising entries, wherein each entry comprises at least one symbol;

receive the uncompressed portion of the data, wherein the uncompressed portion of the data comprises a plurality of symbols;

compare the plurality of symbols with entries in the history table in a parallel fashion, wherein said comparing produces compare results;

determine match information for each of the plurality of symbols based on the compare results; and

output the compressed portion of the data in response to the match information.

33. The data compression system of claim 32, wherein said compressing is performed by the plurality of compression engines in a parallel fashion to produce a plurality of compressed portions of the data.

34. The data compression system of claim 33, further comprising second logic coupled to the plurality of compression engines and configured to:  
receive the plurality of compressed portions of the data from the plurality of compression engines; and  
merge the plurality of compressed portions of the data to produce compressed data.

35. (Amended) A memory controller, comprising:  
memory control logic for controlling a memory; and  
a plurality of parallel compression engines, wherein each of the plurality of parallel compression engines operates independently and implements a lossless parallel data compression algorithm;  
wherein each of the plurality of parallel compression engines is operable to:  
receive a different respective portion of uncompressed data; and  
compress the different respective portion of the uncompressed data using the parallel data compression algorithm to produce a respective compressed portion of the uncompressed data; and

output the respective compressed portion;

wherein the plurality of parallel compression engines are configured to perform said compression in a parallel fashion to produce a plurality of respective compressed portions of the uncompressed data;

wherein the respective compressed portions output from the plurality of parallel compression engines are combinable to form compressed data corresponding to the uncompressed data.

36. The memory controller of claim 35,

wherein, in performing said compression in a parallel fashion, the plurality of parallel compression engines operate concurrently to compress the different respective portions of the uncompressed data to produce the compressed portions of the uncompressed data.

37. The memory controller of claim 35, wherein each of the plurality of parallel compression engines implements a parallel dictionary-based data compression algorithm.

38. The memory controller of claim 35, wherein the uncompressed data comprises a plurality of symbols, wherein each of the plurality of parallel compression engines is operable to compare each of a plurality of received symbols with each of a plurality of entries in a history table concurrently.

39. The memory controller of claim 35,

wherein each of the plurality of parallel compression engines comprises:

an input for receiving the different respective portion of the uncompressed data, wherein the uncompressed data comprises a plurality of symbols, wherein the plurality of symbols includes a first symbol, a last symbol, and one or more middle symbols;

a history table comprising entries, wherein each entry comprises at least one symbol;

a plurality of comparators for comparing the plurality of symbols with entries in the history table, wherein the plurality of comparators are operable to compare each of the plurality of symbols with each entry in the history table concurrently, wherein the plurality of comparators produce compare results;

match information logic coupled to the plurality of comparators for determining match information for each of the plurality of symbols based on the compare results, wherein the match information logic is operable to determine if a contiguous match occurs for one or more of the one or more middle symbols that does not involve a match with either the first symbol or the last symbol; and

an output coupled to the match information logic for outputting compressed data in response to the match information.



40. (Amended) A memory module, comprising:  
one or more memory devices for storing data; and  
a plurality of parallel compression engines, wherein each of the plurality of parallel  
compression engines operates independently and implements a lossless parallel data  
compression algorithm;  
wherein each of the plurality of parallel compression engines is operable to:  
receive a different respective portion of uncompressed data; and  
compress the different respective portion of the uncompressed data using the parallel  
data compression algorithm to produce a respective compressed portion of the  
uncompressed data; and  
output the respective compressed portion;  
wherein the plurality of parallel compression engines are configured to perform said  
compression in a parallel fashion to produce a plurality of respective  
compressed portions of the uncompressed data;  
wherein the respective compressed portions output from the plurality of parallel  
compression engines are combinable to form compressed data corresponding  
to the uncompressed data.

41. The memory module of claim 40,  
wherein, in performing said compression in a parallel fashion, the plurality of parallel compression engines  
operate concurrently to compress the different respective portions of the uncompressed data to produce  
the compressed portions of the uncompressed data.

42. The memory module of claim 40, wherein each of the plurality of parallel compression engines  
implements a parallel dictionary-based data compression algorithm.

43. The memory module of claim 40, wherein the uncompressed data comprises a plurality of symbols,  
wherein each of the plurality of parallel compression engines is operable to compare each of a plurality of received  
symbols with each of a plurality of entries in a history table concurrently.

44. The memory module of claim 40,  
wherein each of the plurality of parallel compression engines comprises:  
an input for receiving the different respective portion of the uncompressed data, wherein the  
uncompressed data comprises a plurality of symbols, wherein the plurality of symbols includes  
a first symbol, a last symbol, and one or more middle symbols;

a history table comprising entries, wherein each entry comprises at least one symbol;  
a plurality of comparators for comparing the plurality of symbols with entries in the history table,  
wherein the plurality of comparators are operable to compare each of the plurality of symbols  
with each entry in the history table concurrently, wherein the plurality of comparators produce  
compare results;  
match information logic coupled to the plurality of comparators for determining match information for  
each of the plurality of symbols based on the compare results, wherein the match information  
logic is operable to determine if a contiguous match occurs for one or more of the one or more  
middle symbols that does not involve a match with either the first symbol or the last symbol;  
and  
an output coupled to the match information logic for outputting compressed data in response to the  
match information.

45. (Amended) A network device, comprising:  
network logic for performing networking functions; and  
a plurality of parallel compression engines, wherein each of the plurality of parallel  
compression engines operates independently and implements a lossless parallel data  
compression algorithm;  
wherein each of the plurality of parallel compression engines is operable to:  
receive a different respective portion of uncompressed data; and  
compress the different respective portion of the uncompressed data using the parallel  
data compression algorithm to produce a respective compressed portion of the  
uncompressed data; and  
output the respective compressed portion;  
wherein the plurality of parallel compression engines are configured to perform said  
compression in a parallel fashion to produce a plurality of respective  
compressed portions of the uncompressed data;  
wherein the respective compressed portions output from the plurality of parallel  
compression engines are combinable to form compressed data corresponding  
to the uncompressed data.

46. The network device of claim 45,  
wherein, in performing said compression in a parallel fashion, the plurality of parallel compression engines  
operate concurrently to compress the different respective portions of the uncompressed data to produce  
the compressed portions of the uncompressed data.

47. The network device of claim 45, wherein each of the plurality of parallel compression engines implements a parallel dictionary-based data compression algorithm.

48. The network device of claim 45, wherein the uncompressed data comprises a plurality of symbols, wherein each of the plurality of parallel compression engines is operable to compare each of a plurality of received symbols with each of a plurality of entries in a history table concurrently.

A<sup>2</sup>

49. The network device of claim 45,  
wherein each of the plurality of parallel compression engines comprises:  
an input for receiving the different respective portion of the uncompressed data, wherein the  
uncompressed data comprises a plurality of symbols, wherein the plurality of symbols includes  
a first symbol, a last symbol, and one or more middle symbols;  
a history table comprising entries, wherein each entry comprises at least one symbol;  
a plurality of comparators for comparing the plurality of symbols with entries in the history table,  
wherein the plurality of comparators are operable to compare each of the plurality of symbols  
with each entry in the history table concurrently, wherein the plurality of comparators produce  
compare results;  
match information logic coupled to the plurality of comparators for determining match information for  
each of the plurality of symbols based on the compare results, wherein the match information  
logic is operable to determine if a contiguous match occurs for one or more of the one or more  
middle symbols that does not involve a match with either the first symbol or the last symbol;  
and  
an output coupled to the match information logic for outputting compressed data in response to the  
match information.

50. (Amended) A data compression system comprising:  
a plurality of compression engines, wherein each of the plurality of compression engines  
operates independently and implements a parallel data compression algorithm;  
first logic coupled to the plurality of compression engines and configured to:  
receive uncompressed data; and  
provide a different portion of the uncompressed data to each of the plurality of  
compression engines;  
wherein each of the plurality of compression engines is configured to:  
compress the uncompressed portion of the uncompressed data provided to the  
particular compression engine to produce a compressed portion of the  
uncompressed data; and

output the compressed portion of the uncompressed data;  
wherein the plurality of compression engines are configured to perform said  
compressing in a parallel fashion to produce a plurality of compressed portions  
of the uncompressed data in parallel; and  
second logic coupled to the plurality of compression engines and configured to:  
receive the plurality of compressed portions of the uncompressed data; and  
combine the plurality of compressed portions of the uncompressed data to produce  
compressed data.

51. The system of claim 50, wherein the system further comprises:  
a processor;  
a memory coupled to the processor and to the second logic and configured to store data for use by the processor;  
wherein the second logic is further configured to write the compressed data to the memory.

A<sup>2</sup>  
52. (Amended) A system comprising:  
a processor;  
a memory coupled to the processor and operable to store data for use by the processor;  
a plurality of compression engines, wherein each of the plurality of compression engines  
operates independently and implements a parallel data compression algorithm; and  
first logic coupled to the memory and to the plurality of compression engines and configured  
to:  
receive uncompressed first data;  
split the uncompressed first data into a plurality of uncompressed portions of the first  
data; and  
provide the plurality of uncompressed portions of the uncompressed first data to the  
plurality of compression engines; and  
wherein the plurality of compression engines are configured to operate concurrently to  
compress the plurality of uncompressed portions of the uncompressed first data to  
produce a plurality of compressed portions of the uncompressed first data.

53. The system of claim 52, further comprising  
second logic coupled to the plurality of compression engines and to the memory and configured to merge the  
plurality of compressed portions of the uncompressed first data to produce compressed first data;  
wherein the second logic is further configured to write the compressed first data to the memory.

54. The system of claim 52, further comprising:  
a plurality of decompression engines;  
third logic coupled to the memory and to the plurality of decompression engines and configured to:  
receive compressed second data;  
split the compressed second data into a plurality of compressed portions of the compressed second data;  
provide the plurality of compressed portions of the compressed second data to the plurality of  
decompression engines; and  
wherein the plurality of decompression engines are configured to operate concurrently to decompress  
the plurality of compressed portions of the compressed second data to produce a plurality of  
uncompressed portions of the compressed second data.

55. The system of claim 54, wherein each of the plurality of decompression engines implements a parallel  
data decompression algorithm

A<sup>2</sup>  
56. The system of claim 54, further comprising:  
fourth logic coupled to the plurality of decompression engines and configured to combine the plurality of  
uncompressed portions of the compressed second data to produce uncompressed second data.

57. (Amended) A method for compressing data, the method comprising:  
receiving uncompressed data;  
providing a different respective portion of the uncompressed data to each of a plurality of  
parallel compression engines, wherein each of the plurality of parallel compression  
engines operates independently and implements a parallel data compression algorithm;  
each of the plurality of parallel compression engines compressing the different respective  
portion of the uncompressed data using the parallel data compression algorithm to  
produce a respective compressed portion of the uncompressed data, wherein the  
plurality of parallel compression engines operate concurrently to perform said  
compressing in a parallel fashion, wherein the plurality of parallel compression  
engines produce a plurality of respective compressed portions of the uncompressed  
data;  
combining the plurality of respective compressed portions of the uncompressed data to  
produce compressed data, wherein the compressed data corresponds to the  
uncompressed data; and  
outputting the compressed data.

58. The method of claim 57, wherein each of the plurality of parallel compression engines implements a parallel lossless data compression algorithm.

59. The method of claim 57, wherein each of the plurality of parallel compression engines implements a parallel statistical data compression algorithm.

60. The method of claim 57, wherein each of the plurality of parallel compression engines implements a parallel dictionary-based data compression algorithm.

61. The method of claim 57, further comprising writing the compressed data to a memory.

62. The method of claim 57,  
wherein, for each of the plurality of parallel compression engines, said compressing comprises:  
receiving the different respective portion of the uncompressed data, wherein the uncompressed data comprises a plurality of symbols, wherein the plurality of symbols includes a first symbol, a last symbol, and one or more middle symbols;  
maintaining a history table comprising entries, wherein each entry comprises at least one symbol;  
comparing the plurality of symbols with entries in the history table in a parallel fashion, wherein said comparing in a parallel fashion comprises comparing each of the plurality of symbols with each entry in the history table concurrently, wherein said comparing produces compare results;  
determining match information for each of the plurality of symbols based on the compare results, wherein said determining match information includes determining if a contiguous match occurs for one or more of the one or more middle symbols that does not involve a match with either the first symbol or the last symbol; and  
outputting compressed data in response to the match information.

63. (Amended) A method comprising:  
receiving uncompressed data;  
providing a different portion of the uncompressed data to each of a plurality of compression engines, wherein each of the plurality of compression engines operates independently and implements a parallel data compression algorithm;  
each of the plurality of compression engines compressing its respective different portion of the uncompressed data to produce a compressed portion of the data, wherein said compressing comprises:  
maintaining a history table comprising entries, wherein each entry comprises at least one symbol;

receiving the respective different portion of the uncompressed data, wherein the  
respective different portion of the uncompressed data comprises a plurality of  
symbols;  
comparing the plurality of symbols with entries in the history table in a parallel  
fashion, wherein said comparing produces compare results;  
determining match information for each of the plurality of symbols based on the  
compare results; and  
outputting the compressed portion of the data in response to the match information.  
wherein said compressing is performed by the plurality of compression engines in a parallel  
fashion to produce a plurality of compressed portions of the uncompressed data.

64. The method of claim 63, further comprising:  
merging the plurality of compressed portions of the uncompressed data to produce compressed data; and  
writing the compressed data to a memory.

65. (Amended) A data decompression system comprising:  
a plurality of parallel decompression engines, wherein each of the plurality of parallel  
decompression engines operates independently and implements a parallel data  
decompression algorithm;  
wherein each of the plurality of parallel decompression engines is operable to:  
receive a different respective portion of compressed data; and  
decompress the different respective portion of the compressed data using the parallel  
data decompression algorithm to produce a respective uncompressed portion of  
the compressed data; and  
output the respective uncompressed portion;  
wherein the plurality of parallel decompression engines are configured to perform said  
decompression in a parallel fashion to produce a plurality of respective  
uncompressed portions of the compressed data.

66. The data decompression system of claim 65,  
wherein, in performing said decompression in a parallel fashion, the plurality of parallel decompression engines  
operate concurrently to decompress the different respective portions of the compressed data to produce  
the uncompressed portions of the compressed data.

- A<sup>2</sup>
67. The data decompression system of claim 65, wherein the respective uncompressed portions output from the plurality of parallel decompression engines are combinable to form uncompressed data corresponding to the compressed data.
68. The data decompression system of claim 65, wherein each of the plurality of parallel decompression engines implements a parallel lossless data decompression algorithm.
69. The data decompression system of claim 65, wherein each of the plurality of parallel decompression engines implements a parallel statistical data decompression algorithm.
70. The data compression system of claim 65, wherein each of the plurality of parallel decompression engines implements a parallel dictionary-based data decompression algorithm.
71. The data compression system of claim 70, wherein each of the plurality of parallel decompression engines implements a parallel data decompression algorithm based on a Lempel-Ziv (LZ) algorithm.
72. The data decompression system of claim 70, wherein the compressed data comprises a compressed representation of uncompressed data, wherein the uncompressed data has a plurality of symbols; wherein, in decompressing the different respective portion of the compressed data, each of the plurality of parallel decompression engines is operable to:  
receive the different respective portion of the compressed data, wherein the different respective portion of the compressed data comprises tokens each describing one or more of the symbols in the uncompressed data;  
examine a plurality of tokens from the different respective portion of the compressed data in parallel in a current decompression cycle; and  
generate the uncompressed data comprising the plurality of symbols in response to said examining.
73. The data decompression system of claim 72, wherein, in examining the plurality of tokens from the different respective portion of the compressed data in parallel, each of the plurality of parallel decompression engines is operable to operate on the plurality of tokens concurrently.
74. The data decompression system of 73, wherein each of the plurality of parallel decompression engines operates in a pipelined fashion; wherein, in examining the plurality of tokens from the different respective portion of the compressed data in parallel, each of the plurality of parallel decompression engines is operable to operate on the plurality of tokens during a single pipeline stage.



75. The data decompression system of claim 72, wherein each of the plurality of parallel decompression engines is further operable to:

generate a plurality of selects in parallel in response to examining the plurality of tokens in parallel, wherein each of the plurality of selects points to a symbol in a combined history window;  
wherein each of the plurality of parallel decompression engines generates the uncompressed data using the plurality of selects.

76. (Amended) A data decompression system comprising:  
a plurality of decompression engines, wherein each of the plurality of decompression engines operates independently and implements a parallel data decompression algorithm;  
first logic coupled to the plurality of decompression engines and configured to:  
receive compressed data; and  
provide a different respective portion of the compressed data to each of the plurality of decompression engines;

wherein each of the plurality of decompression engines is configured to:  
decompress the respective compressed portion of the compressed data to produce an uncompressed portion of the compressed data; and  
output the uncompressed portion of the compressed data;  
wherein the plurality of decompression engines are configured to operate concurrently to perform said decompressing in a parallel fashion to produce a plurality of uncompressed portions of the compressed data.

77. The data decompression system of claim 76, further comprising:  
second logic coupled to the plurality of decompression engines and configured to:  
receive the plurality of uncompressed portions of the compressed data; and  
merge the plurality of uncompressed portions of the compressed data to produce uncompressed data.

78. The data decompression system of claim 76, wherein each of the plurality of parallel decompression engines implements a parallel lossless data decompression algorithm.

79. The data decompression system of claim 76, wherein each of the plurality of parallel decompression engines implements a parallel statistical data decompression algorithm.

80. The data compression system of claim 76, wherein each of the plurality of parallel decompression engines implements a parallel dictionary-based data decompression algorithm.

81. The data decompression system of claim 76, wherein the parallel data decompression algorithm is based on one of an LZSS algorithm, an LZ77 algorithm, an LZ78 algorithm, an LZW algorithm, an LZRW1 algorithm, a Run Length Encoding (RLE) algorithm, a Predictive Encoding algorithm, a Huffman coding algorithm, an Arithmetic coding algorithm and a Differential decompression algorithm.

82. (Amended) A data decompression system comprising:  
a plurality of decompression engines, wherein each of the plurality of decompression engines operates independently and implements a parallel data decompression algorithm;  
first logic coupled to the plurality of decompression engines and configured to:  
receive compressed data;  
provide a different portion of the compressed data to each of the plurality of decompression engines;  
wherein each of the plurality of decompression engines is configured to decompress its received different portion of the compressed data to produce an uncompressed portion of the data, wherein, in said decompressing, each of the plurality of decompression engines is configured to:  
receive the different portion of the compressed data, wherein the different portion of the compressed data comprises tokens each describing one or more uncompressed symbols;  
examine a plurality of tokens from the different portion of the compressed data in parallel in a current decompression cycle;  
generate a plurality of selects in parallel in response to examining the plurality of tokens in parallel, wherein each of the plurality of selects points to a symbol in a combined history window; and  
generate an uncompressed portion of the compressed data comprising the plurality of symbols using the plurality of selects.

83. The data decompression system of claim 82, wherein said decompressing is performed by the plurality of decompression engines in a parallel fashion to produce a plurality of uncompressed portions of the compressed data.

84. The data decompression system of claim 83, further comprising  
second logic coupled to the plurality of decompression engines and configured to:  
receive the plurality of uncompressed portions of the compressed data from the plurality of decompression engines; and

merge the plurality of uncompressed portions of the compressed data to produce uncompressed data.

85. (Amended) A data decompression system comprising:

a plurality of decompression engines, wherein each of the plurality of decompression engines operates independently and implements a parallel data decompression algorithm;

first logic coupled to the plurality of decompression engines and configured to:

receive compressed data; and

provide a different portion of the compressed data to each of the plurality of decompression engines;

wherein each of the plurality of decompression engines is configured to:

decompress the compressed portion of the data provided to the particular

decompression engine to produce an uncompressed portion of the data; and

output the uncompressed portion of the data;

wherein the plurality of decompression engines is configured to perform said

decompressing in a parallel fashion to produce a plurality of uncompressed portions of the data in parallel; and

second logic coupled to the plurality of decompression engines and configured to:

receive the plurality of uncompressed portions of the data; and

merge the plurality of uncompressed portions of the data to produce uncompressed data.

86. (Amended) A memory controller, comprising:

memory control logic for controlling a memory; and

a plurality of parallel decompression engines, wherein each of the plurality of parallel decompression engines operates independently and implements a parallel data decompression algorithm;

wherein each of the plurality of parallel decompression engines is operable to:

receive a different respective portion of compressed data; and

decompress the different respective portion of the compressed data using the parallel

data decompression algorithm to produce a respective uncompressed portion of the compressed data; and

output the respective uncompressed portion;

wherein the plurality of parallel decompression engines are configured to perform said decompression in a parallel fashion to produce a plurality of respective uncompressed portions of the compressed data;  
wherein the respective uncompressed portions output from the plurality of parallel decompression engines are combinable to form uncompressed data corresponding to the compressed data.

87. The memory controller of claim 86,  
wherein, in performing said decompression in a parallel fashion, the plurality of parallel decompression engines operate concurrently to decompress the different respective portions of the compressed data to produce the uncompressed portions of the compressed data.

88. The memory controller of claim 86,  
wherein each of the plurality of parallel decompression engines implements a parallel dictionary-based data decompression algorithm.

89. The memory controller of claim 88, wherein the compressed data comprises a compressed representation of uncompressed data, wherein the uncompressed data has a plurality of symbols;  
wherein, in decompressing the different respective portion of the compressed data, each of the plurality of parallel decompression engines is operable to:  
receive the compressed data, wherein the compressed data comprises tokens each describing one or more of the symbols in the uncompressed data;  
examine a plurality of tokens from the compressed data in parallel in a current decompression cycle;  
and  
generate the uncompressed data comprising the plurality of symbols in response to said examining.

90. The memory controller of claim 89,  
wherein, in examining the plurality of tokens from the compressed data in parallel, each of the plurality of parallel decompression engines is operable to operate on the plurality of tokens concurrently.

91. The memory controller of 90,  
wherein each of the plurality of parallel decompression engines operates in a pipelined fashion;  
wherein, in examining the plurality of tokens from the compressed data in parallel, each of the plurality of parallel decompression engines is operable to operate on the plurality of tokens during a single pipeline stage.

92. The memory controller of claim 89, wherein each of the plurality of parallel decompression engines is further operable to:

generate a plurality of selects in parallel in response to examining the plurality of tokens in parallel, wherein each of the plurality of selects points to a symbol in a combined history window;  
wherein each of the plurality of parallel decompression engines generates the uncompressed data using the plurality of selects.

93. (Amended) A memory module, comprising:

at least one memory device for storing data; and

A2 a plurality of parallel decompression engines, wherein each of the plurality of parallel decompression engines operates independently and implements a parallel data decompression algorithm;

wherein each of the plurality of parallel decompression engines is operable to:

receive a different respective portion of compressed data; and

decompress the different respective portion of the compressed data using the parallel data decompression algorithm to produce a respective uncompressed portion of the compressed data; and

output the respective uncompressed portion;

wherein the plurality of parallel decompression engines are configured to perform said decompression in a parallel fashion to produce a plurality of respective uncompressed portions of the compressed data;

wherein the respective uncompressed portions output from the plurality of parallel decompression engines are combinable to form uncompressed data corresponding to the compressed data.

94. The memory module of claim 93,

wherein, in performing said decompression in a parallel fashion, the plurality of parallel decompression engines operate concurrently to decompress the different respective portions of the compressed data to produce the uncompressed portions of the compressed data.

95. The memory module of claim 93,

wherein each of the plurality of parallel decompression engines implements a parallel dictionary-based data decompression algorithm.

96. The memory module of claim 95, wherein the compressed data comprises a compressed representation of uncompressed data, wherein the uncompressed data has a plurality of symbols;

wherein, in decompressing the different respective portion of the compressed data, each of the plurality of parallel decomposition engines is operable to:

receive the compressed data, wherein the compressed data comprises tokens each describing one or more of the symbols in the uncompressed data;

examine a plurality of tokens from the compressed data in parallel in a current decompression cycle; and

generate the uncompressed data comprising the plurality of symbols in response to said examining.

97. The memory controller of claim 96,

wherein, in examining the plurality of tokens from the compressed data in parallel, each of the plurality of parallel decomposition engines is operable to operate on the plurality of tokens concurrently.

98. The memory controller of 97,

wherein each of the plurality of parallel decomposition engines operates in a pipelined fashion;

wherein, in examining the plurality of tokens from the compressed data in parallel, each of the plurality of parallel decomposition engines is operable to operate on the plurality of tokens during a single pipeline stage.

99. The memory controller of claim 96, wherein each of the plurality of parallel decomposition engines is further operable to:

generate a plurality of selects in parallel in response to examining the plurality of tokens in parallel, wherein each of the plurality of selects points to a symbol in a combined history window;

wherein each of the plurality of parallel decomposition engines generates the uncompressed data using the plurality of selects.

100. (Amended) A network device, comprising:

network logic for interfacing to a network; and

a plurality of parallel decomposition engines, wherein each of the plurality of parallel decomposition engines operates independently and implements a parallel data decomposition algorithm;

wherein each of the plurality of parallel decomposition engines is operable to:

receive a different respective portion of compressed data; and

decompress the different respective portion of the compressed data using the parallel data decomposition algorithm to produce a respective uncompressed portion of the compressed data; and

output the respective uncompressed portion;  
wherein the plurality of parallel decompression engines are configured to perform said decompression in a parallel fashion to produce a plurality of respective uncompressed portions of the compressed data;  
wherein the respective uncompressed portions output from the plurality of parallel decompression engines are combinable to form uncompressed data corresponding to the compressed data.

101. The network device of claim 100,  
wherein, in performing said decompression in a parallel fashion, the plurality of parallel decompression engines operate concurrently to decompress the different respective portions of the compressed data to produce the uncompressed portions of the compressed data.

102. The network device of claim 100,  
wherein each of the plurality of parallel decompression engines implements a parallel dictionary-based data decompression algorithm.

103. The network device of claim 102, wherein the compressed data comprises a compressed representation of uncompressed data, wherein the uncompressed data has a plurality of symbols;  
wherein, in decompressing the different respective portion of the compressed data, each of the plurality of parallel decompression engines is operable to:  
receive the compressed data, wherein the compressed data comprises tokens each describing one or more of the symbols in the uncompressed data;  
examine a plurality of tokens from the compressed data in parallel in a current decompression cycle;  
and  
generate the uncompressed data comprising the plurality of symbols in response to said examining.

104. The network device of claim 103,  
wherein, in examining the plurality of tokens from the compressed data in parallel, each of the plurality of parallel decompression engines is operable to operate on the plurality of tokens concurrently.

105. The network device of 104,  
wherein each of the plurality of parallel decompression engines operates in a pipelined fashion;  
wherein, in examining the plurality of tokens from the compressed data in parallel, each of the plurality of parallel decompression engines is operable to operate on the plurality of tokens during a single pipeline stage.

106. The network device of claim 103, wherein each of the plurality of parallel decompression engines is further operable to:

generate a plurality of selects in parallel in response to examining the plurality of tokens in parallel, wherein each of the plurality of selects points to a symbol in a combined history window;  
wherein each of the plurality of parallel decompression engines generates the uncompressed data using the plurality of selects.

107. (Amended) A method for decompressing data, comprising:

receiving compressed data;

providing a different portion of the compressed data to each of a plurality of decompression engines, wherein each of the plurality of decompression engines operates independently and implements a parallel data decompression algorithm;

each of the plurality of decompression engines decompressing the different portion of the compressed data, wherein said decompressing produces an uncompressed portion of the data, wherein said decompressing is performed by the plurality of decompression engines in a parallel fashion to produce a plurality of uncompressed portions of the compressed data; and

combining the plurality of uncompressed portions of the compressed data to produce uncompressed data.

108. The method of claim 107, further comprising writing the uncompressed data to a memory.

109. The method of claim 107, wherein the parallel data decompression algorithm is a parallel dictionary-based decompression algorithm.

110. The method of claim 107,

wherein the compressed data comprises a compressed representation of uncompressed data, wherein the uncompressed data has a plurality of symbols;

wherein each of the plurality of decompression engines decompressing the different portion of the compressed data comprises;

receiving the different portion of the compressed data, wherein the compressed data comprises tokens each describing one or more of the symbols in the uncompressed data;

examining a plurality of tokens from the compressed data in parallel in a current decompression cycle; and  
generating the uncompressed data comprising the plurality of symbols in response to said examining.



111. (Amended) A method comprising:  
receiving compressed data;  
providing a different portion of the compressed data to each of a plurality of decompression engines, wherein each of the plurality of decompression engines operates independently and implements a parallel data decompression algorithm;  
each of the plurality of decompression engines decompressing a compressed portion of the data provided to the particular decompression engine to produce an uncompressed portion of the data, wherein said decompressing comprises:  
receiving the compressed portion of the data, wherein the compressed portion of the data comprises tokens each describing one or more uncompressed symbols;  
examining a plurality of tokens from the compressed portion of the data in parallel in a current decompression cycle;  
generating a plurality of selects in parallel in response to examining the plurality of tokens in parallel, wherein each of the plurality of selects points to a symbol in a combined history window; and  
generating an uncompressed portion of the data comprising the plurality of symbols using the plurality of selects;  
wherein said decompressing is performed by the plurality of decompression engines in a parallel fashion to produce a plurality of uncompressed portions of the data.

112. The method of claim 111, further comprising:  
merging the plurality of uncompressed portions of the data to produce uncompressed data; and  
writing the uncompressed data to a memory.

113. (Amended) A data compression/decompression system comprising:  
a plurality of compression engines, wherein each of the plurality of compression engines operates independently and implements a parallel data compression algorithm;  
a plurality of decompression engines, wherein each of the plurality of decompression engines implements a parallel data decompression algorithm;  
first logic coupled to the plurality of data compression engines and to the plurality of data decompression engines and configured to:  
receive data;

if the data is uncompressed, provide a plurality of uncompressed portions of the data to each of the plurality of data compression engines; and  
if the data is compressed, provide a plurality of compressed portions of the data to each of the plurality of data decompression engines;  
wherein, if the data is uncompressed, the plurality of compression engines are configured to compress the plurality of uncompressed portions of the data in a parallel fashion to produce a plurality of compressed portions of the data; and  
wherein, if the data is compressed, the plurality of decompression engines are configured to decompress the plurality of compressed portions of the data in a parallel fashion to produce a plurality of uncompressed portions of the data.

A<sup>2</sup>  
114. The data compression/decompression system of claim 113, further comprising:  
second logic coupled to the plurality of data compression engines and to the plurality of data decompression engines and configured to:  
if the data is uncompressed, merge the compressed portions of the data produced by the plurality of compression engines to produced compressed data; and  
if the data is compressed, merge the uncompressed portions of the data produced by the plurality of decompression engines to produced uncompressed data.

115. The data compression/decompression system of claim 113, wherein the parallel data decompression algorithm and the parallel data decompression algorithm are based on a serial lossless data compression/decompression algorithm.

116. (Amended) A data compression/decompression system comprising:  
a plurality of compression/decompression engines, wherein each of the plurality of compression/decompression engines operates independently and implements a parallel data compression algorithm and a parallel data decompression algorithm;  
first logic coupled to the plurality of data compression/decompression engines and configured to:  
receive data;  
split the data into a plurality of portions of the data; and  
provide the plurality of portions of the data to the plurality of data compression/decompression engines;  
wherein the plurality of data compression/decompression engines is configured to:

if the data is uncompressed, compress the portions of the data in a parallel fashion to produce a plurality of compressed portions of the first data; and  
if the data is compressed, decompress the portions of the data in a parallel fashion to produce a plurality of uncompressed portions of the first data.

117. The data compression/decompression system of claim 116, further comprising:  
second logic coupled to the plurality of data compression/decompression engines and to the plurality of data decompression engines and configured to  
if the data is uncompressed, merge the compressed portions of the data produced by the plurality of compression/decompression engines to produced compressed data; and  
if the data is compressed, merge the uncompressed portions of the data produced by the plurality of compression/decompression engines to produced uncompressed data.

118. The data compression/decompression system of claim 116, wherein the parallel data decompression algorithm and the parallel data decompression algorithm are lossless parallel dictionary-based compression/decompression algorithms.

119. (Amended) A system comprising:  
a processor;  
a memory coupled to the processor and operable to store data for use by the processor;  
a data compression/decompression system comprising:  
a plurality of compression engines, wherein each of the plurality of compression engines operates independently and implements a parallel data compression algorithm;  
a plurality of decompression engines, wherein each of the plurality of decompression engines implements a parallel data decompression algorithm;  
first logic coupled to the plurality of data compression engines and to the plurality of data decompression engines and configured to:  
receive first data;  
if the first data is uncompressed, provide a plurality of uncompressed portions of the first data to each of the plurality of compression engines; and  
if the first data is compressed, provide a plurality of compressed portions of the first data to each of the plurality of decompression engines;  
wherein, if the first data is uncompressed, the plurality of compression engines is configured to compress the plurality of uncompressed portions of the first data

in a parallel fashion to produce a plurality of compressed portions of the first data; and

wherein, if the first data is compressed, the plurality of decompression engines is configured to decompress the plurality of compressed portions of the first data in a parallel fashion to produce a plurality of uncompressed portions of the first data.

A2  
10/044